What Is Claimed Is:

1. A method for executing a software update of a control unit by flash programming a flash memory of the control unit having multiple segments, via a serial interface comprising at least the following steps:

a) establishing the requirements to be set for the flash programming;

b) establishing a flash programming sequence, using a finite-state machine which defines states and transitions of the software;

c) checking the availability, security, and reliability requirements of each state and each transition of the finite-state machine.

2. The method as recited in Claim 1, wherein different operating states, in particular a "starting state," a "normal state," and a "software update," transitions between the operating states, and transition conditions are specified for the software of the control unit.

3. A method as recited in Claim 1 or 2, wherein memory arrays of the software of the control unit, which are relevant for flash programming, are divided into programmable and non-programmable memory arrays, and software components to be reprogrammed are correspondingly assigned to the memory arrays.

4. The method as recited in Claim 3, wherein the memory arrays of the software are each assigned to a memory of the control unit, in particular one programmable

memory array to at least one segment of the flash memory and one non-programmable memory array to a ROM of the control unit.

5. The method as recited in one of the preceding claims, wherein a boot block, which provides software functionality necessary for executing the flash programming procedure, a program stock and a data stock are stored in segments of the flash memory of the control unit, and a start-up block, which provides software functionality necessary for executing the flash programming procedure, is stored in a ROM of the control unit.

6. The method as recited in one of the preceding claims, wherein security, reliability and availability requirements of the flash programming procedure are specified.

7. The method as recited in one of Claims 2 through 6, wherein, substates, transitions between them, and transition conditions, adoptable in the "software update" operating state, are specified by the finite-state machine during execution of the flash programming procedure.

8. A finite-state machine for executing a software update of a control unit by flash programming, which defines all substates of the software of the control unit, transitions between them, and transition conditions adoptable during execution of the software update, and which specifies permanent, non-erasable storage of a last-valid state or an error-free run state in response to the occurrence of a fault during execution of the software update.

9. The finite-state machine as recited in Claim 8, wherein substates for authentication and signature check and

substates for the deletion and programming of segments of the flash memory are specified as "abort/error message" and "completion/success message" substates.

10. A computer program comprising program code elements via which predefined availability, security, and reliability requirements of each state and each transition of a finite-state machine as recited in Claim 8 or 9 are automatically checked when the program code elements are executed on a computer or on a computer system.

11. A method for executing flash programming of a boot block which provides software functionality necessary for executing the flash programming procedure and is stored in a first segment (flash segment A) of a flash memory,
the method comprising at least the following steps:

a) copying the old boot block to be reprogrammed into a free section of a second memory (RAM);

b) activating the old boot block in the second memory (RAM) and deactivating the old boot block in the flash memory;

c) temporarily storing a new boot block in a second segment (flash segment C) of the flash memory;

d) programming the new boot block by copying the second segment (flash segment C) into the first segment (flash segment A);

e) activating the new boot block in the first segment (flash segment A) and deactivating the old boot block in the second memory (RAM).

12. The method as recited in Claim 11, wherein one boot block is always marked in the flash memory as a valid boot block for restarting the flash programming procedure.